How to recover an incorrect application which may have lost the initial values of Function Blocks (EFB and DFB) - PRB-222024.

v1, October 18, 2019

If you have any questions regarding this Technical Application Note, contact your regional support office at https://www.schneider-electric.com/en/work/support

Content

1 Introduction

This procedure describes how to recover an incorrect application which may have lost the initial values of Function Blocks (EFB and DFB).

This procedure applies to all the applications created or modified, using Control Expert V14.0 and running in the M580 Safety offer, following the specific use cases described in the PRB-222024.

This procedure must be followed step by step in order to recover the expected initial values of all DFB / EFB. The DFB created by the user must be diagnosed and, if needed, fixed by the user himself.

Note: Before applying this procedure the installation of ControlExpert_V140_HF2 (or upgrade to Control Expert V14.1) is strongly recommended. Otherwise, the application can become incorrect again when the use cases described in the PRB occur.

2 Technical Context

Context: When a Function Block (FB) is instantiated in an application, the FB type is copied from Control Expert (CE) libset to the application, this element gets corrupted by the incorrect behavior of the software (known as “loss of initial values of internal variables”).

Therefore, three different types of functional blocks can be impacted by the event:

- EFB (elementary Function Blocks). The EFB type is a copy from the CE libset, it may be corrupted and cannot be edited by the user. Recover is described in chapter 3.
- DFB (derived Function Blocks). The DFB type is a copy from the CE libset, may be corrupted and cannot be edited by the user. Recover is described in chapter 4.
- User DFB. This DFB type is created by the user and the user only can diagnose it and if needed, fix it. Recover is described in chapter 5.

Trigger: Building a safety application that was open from an .sta file (archive format) or uploaded from a processor leads to lose some initial values of the internal variables inside the function blocks.

Diagnostic: Users cannot diagnose if an application is now incorrect or not.

Solution: The software bug has been corrected in the ControlExpert_V140_HF2 and starting from Control Expert 14.1. However, it cannot fix an incorrect application.

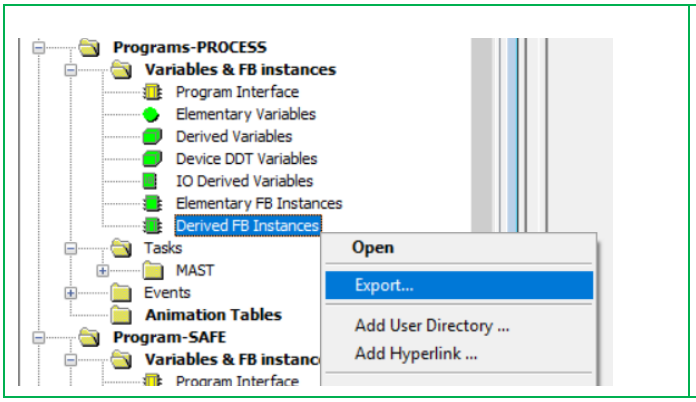
To do so, users are recommended to follow the procedure described below.

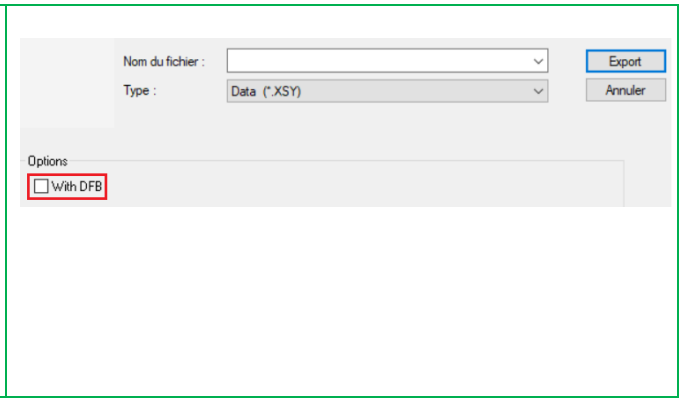
3 Cleanup sequence of the DFB from Control Expert

1. Open user application.

Process Part

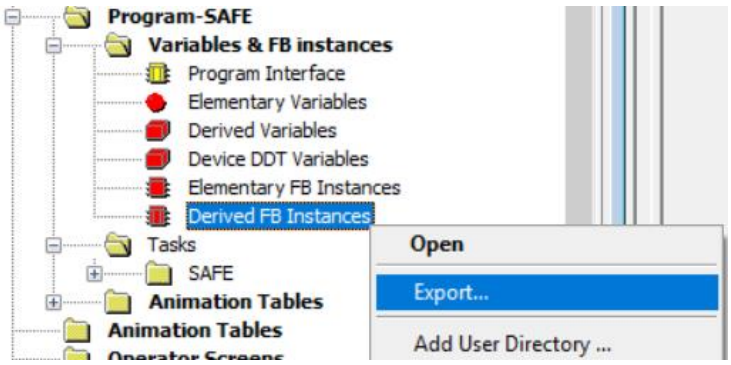
2. Export "Derived FB Instances" for Process part. Make sure to uncheck the dialog box "With DFB" --> process_dfb_instances.xsy .





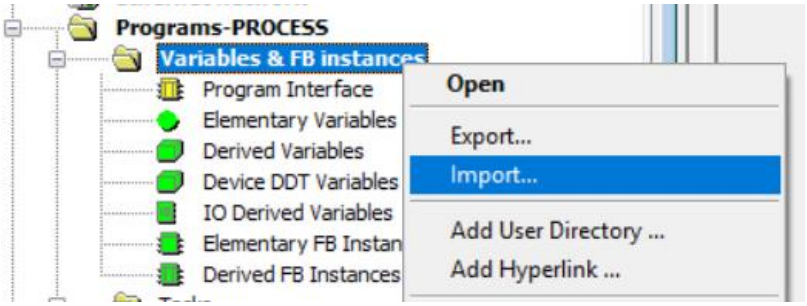
Safe Part

3. Export "Derived FB Instances" for Safe part. Make sure to uncheck the dialog box "With DFB" -->-safe_dfb_instances.xsy.

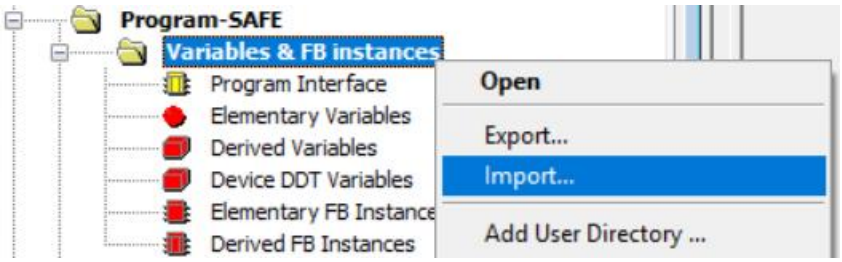


4. Create a new application with the same CPU reference.

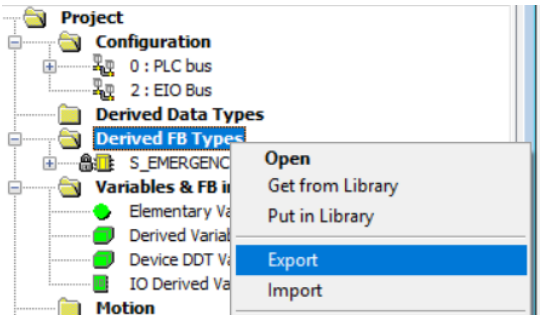
5. Import both files in each dedicated part.
   a. import process_dfb_instances.xsy



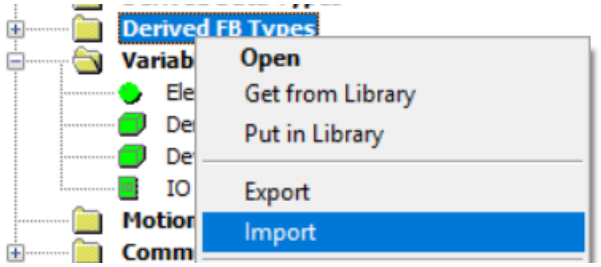
b. import safe_dfb_instances.xsy

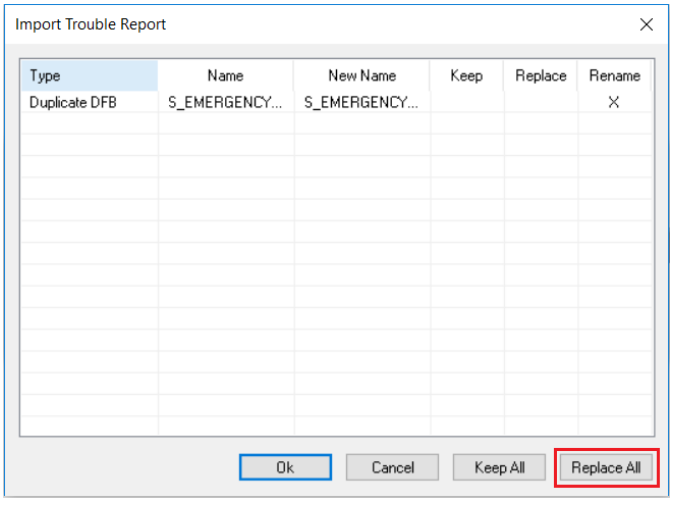


6. Export DFBs -> DFB_types.xdb



7. Close Control Expert.
8. Open Control Expert.
9. Open Customer application.
10. Import all DFBs. Select Replace ALL.





11. Build all and save the application.

4 Cleanup sequence of the EFB from Control Expert

The .zef format doesn't contain the EFB code. Therefore, when opening an application from a .zef file, all EFB types will be copied from the CE libset.

1. Open the user application.
2. Export application in .zef format.
3. Close the application.
4. Open .zef format application.
5. Build all and save the application.

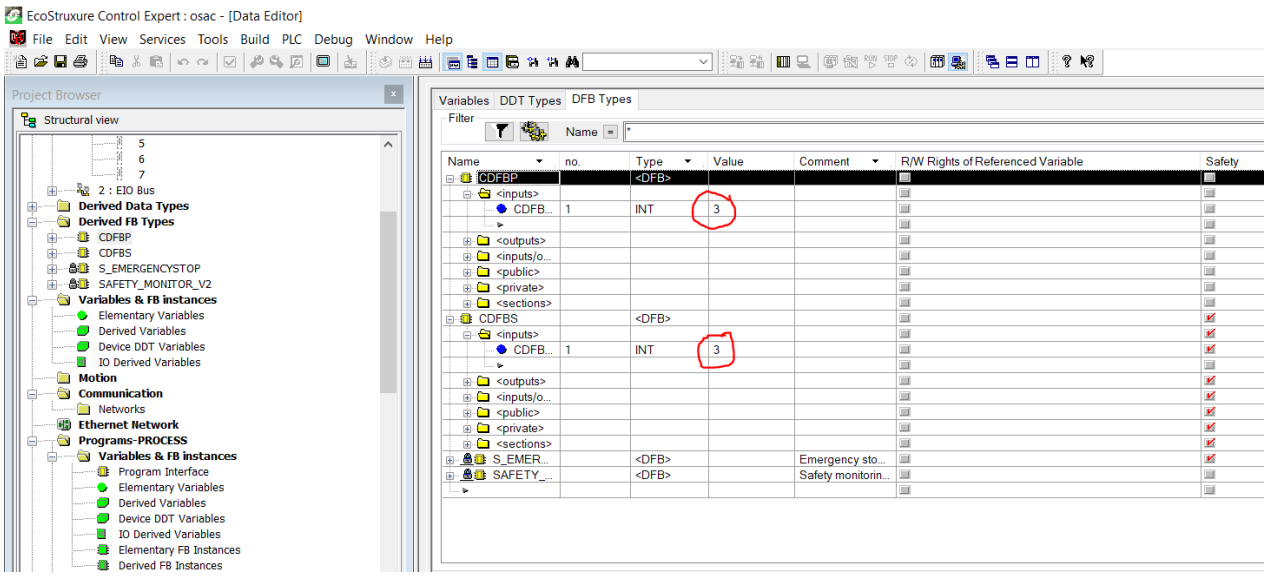
5 Cleanup sequence of the user DFB

Following the event, the initial values of the variables have disappeared. To recover, two different solutions are proposed.
You can choose the option 1a or 1b. First, edit the DFB and check if initial values are present, if not:

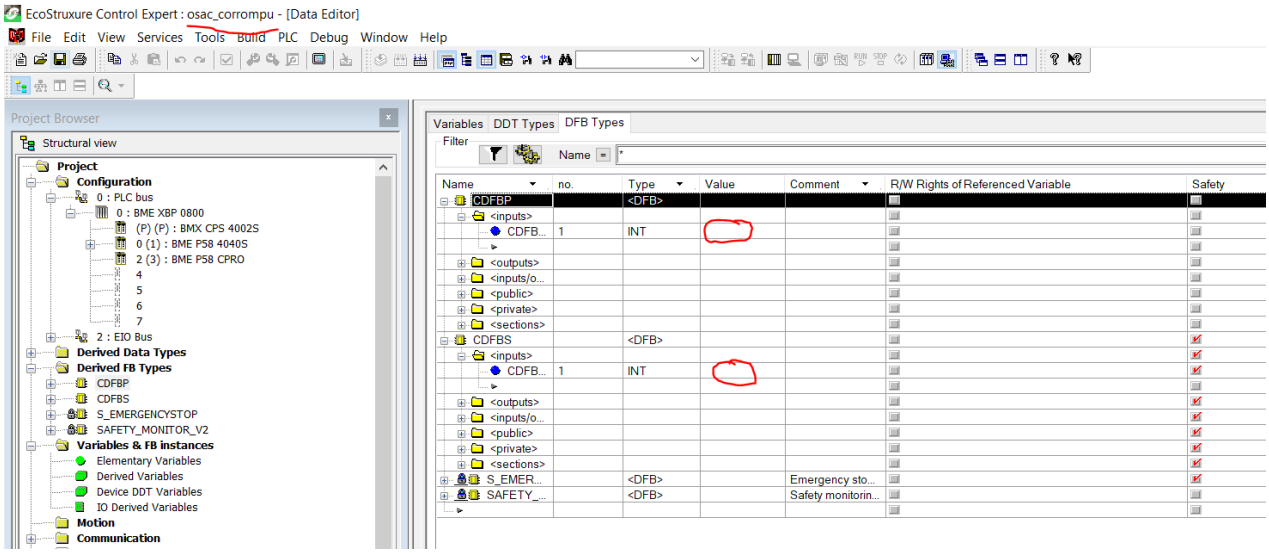
1a. type them again.

1b. import the DFB type from a known correct application.

2. After one of the two options, do the "build all" and save the application.



example of a correct user DFB containing its initial values



example of a corrupted user DFB missing its initial values

6 Conclusion

After all these steps (chapter 3, 4 and 5) the 3 types of function blocks are restored. Please verify the correct behavior of this restored application.